AD-A017 436

APPLICATION OF TACTICAL DATA SYSTEMS FOR TRAINING:
DEVTOS FEASIBILITY DETERMINATION AND SELECTION OF AN
INSTRUCTIONAL OPERATING SYSTEM

William G. Hoyt, et al

System Development Corporation

Prepared for:

Army Research Institute for the Behavioral and Social
Sciences
Deputy Chief of Staff for Personnel (Army)
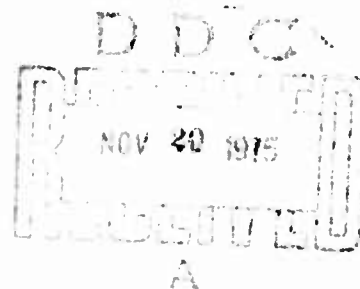
October 1975

Technical Paper 267

AD

328174

APPLICATION OF TACTICAL DATA SYSTEMS FOR TRAINING: DEVTOS FEASIBILITY DETERMINATION AND SELECTION OF AN INSTRUCTIONAL OPERATING SYSTEM

William G. Hoyt, Alfred K. Butler, and Fred D. Bennik
System Development Corporation

SYSTEMS INTEGRATION & COMMAND/CONTROL TECHNICAL AREA

ADA017436

DDC

NOV 20 1975

A

U. S. Army

Research Institute for the Behavioral and Social Sciences

October 1975

U. S. ARMY RESEARCH INSTITUTE

FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency under the Jurisdiction of the
Deputy Chief of Staff for Personnel

J. E. UHLANER
Technical Director

W. C. MAUS
COL, GS
Commander

Research accomplished
under contract to the Department of the Army

System Development Corporation

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Paper 267	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* APPLICATION OF TACTICAL DATA SYSTEMS FOR TRAINING: DEVTOS FEASIBILITY DETERMINATION AND SELECTION OF AN INSTRUCTIONAL OPERATING SYSTEM		5. TYPE OF REPORT & PERIOD COVERED Interim - Dec 72 to Dec 73
		6. PERFORMING ORG. REPORT NUMBER None
7. AUTHOR(s) William G. Hoyt, Alfred K. Butler, and Fred D. Bennik		8. CONTRACT OR GRANT NUMBER(s) Contract DAHC-19-73-C-0029
9. PERFORMING ORGANIZATION NAME AND ADDRESS System Development Corporation, Santa Monica, CA		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2Q062106A721 CS-C-16
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy Chief of Staff for Personnel, Department of the Army		12. REPORT DATE October 1975
		13. NUMBER OF PAGES 62
14. MONITORING AGENCY NAME & ADDRESS(*If different from Controlling Office*) U.S. Army Research Institute for the Behavioral and Social Sciences, Arlington, VA 22209		15. SECURITY CLASS. *(of this report)* Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT *(of this Report)* Approved for public release, distribution unlimited.		
17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, If different from Report)*		
18. SUPPLEMENTARY NOTES First of several planned reports dealing with special aspects of MASSTER Test FM 122, IBCS (Integrated Battlefield Command System): Automated Instruction		
19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)* Computer-Assisted Instruction Computerized Testing PLANIT Author Language Army Tactical Data Systems		Field Test and Evaluation Military Training

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

    The U.S. Army is likely to have a considerable data processing capability at the tactical unit level soon. A potential secondary role for these tactical data systems, when they are not required for tactical operations, is that of supporting unit and individual training. This report is the first of a series on the use of tactical data systems for training.

20. Research reported here established the feasibility of this use. An analysis of the hardware and software of the existing Developmental Tactical Operations System (DEVTOS) at Fort Hood, Texas determined that the DEVTOS could support computer-aided instruction (CAI). A survey and analysis of 23 existing CAI programs in relation to the requirements of DEVTOS clearly identified PLANIT (Programming Language for Interactive Teaching) as the most appropriate. PLANIT consists of an author language and supporting computer programs for preparing, editing, and presenting to students any subject matter suitable for individualized presentation. It provides a wide range of authoring conveniences, a comprehensive calculation utility that allows both author and student to use a natural language to access the computer, necessary support features for time-shared lesson presentation, and automatic data collection and maintenance of progress records from session to session. Maximum portability, a prime objective during PLANIT's development, permits it to be installed on a variety of different computers with relative ease.

la

APPLICATION OF TACTICAL DATA SYSTEMS FOR TRAINING:

DEVTOS FEASIBILITY DETERMINATION AND SELECTION OF AN INSTRUCTIONAL OPERATING SYSTEM

William G. Hoyt, Alfred K. Butler, and Fred D. Bennik
System Development Corporation

James D. Baker, Supervisory Project Director

SYSTEMS INTEGRATION & COMMAND/CONTROL TECHNICAL AREA

Cecil D. Johnson, Chief

Submitted By.
Joseph Zeidner, Director
ORGANIZATIONS & SYSTEMS
RESEARCH LABORATORY

Approved By:
J. E. Uhlaner
TECHNICAL DIRECTOR

U. S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES

Office, Deputy Chief of Staff for Personnel
Department of the Army
1300 Wilson Boulevard, Arlington, Virginia 22209

October 1975

iii

FOREWORD

The Systems Integration and Command/Control Technical Area of the U.S. Army Research Institute for the Behavioral and Social Sciences (ARI) is concerned with research designed to help commanders and staff in the critical functions of assimilating information and making appropriate decisions, to develop techniques for efficient processing and use of information by operational personnel in tactical situations, and to maximize effectiveness of Command and Control systems through the most efficient use of human abilities. Present Army tactical data systems (e.g., ARTADS, IBCS) have been developed out of this concern.

A specific research project under the Command Systems program is designed to optimize the use of Army tactical data systems by developing compatible computer-assisted instruction (CAI) packages that use the data systems to support individual and unit training requirements when the systems are not required for tactical operations. This Technical Paper is the first of several reports to come from the project; a preliminary version has been informally printed as ARI Research Memorandum 74-8. The entire research effort was begun under RDTE Project 2Q062I06A72I and is responsive to requirements of RDTE Project 2Q763731A734, FY I975 Work Program, and to special requirements, originally from the Assistant Chief of Staff for Force Development and the Director of Army Research, now from the Army Training and Doctrine Command and the Project Manager, Computerized Training System (PM CTS).

ARI research in this area is conducted as an in-house effort augmented by contracts with organizations selected for their unique capabilities for research in the area. The present study was conducted jointly by personnel of ARI and the System Development Corporation, with special contributions by personnel listed in the section "Sources of Information."

J. E. UHLANER
Technical Director

iv

APPLICATION OF TACTICAL DATA SYSTEMS FOR TRAINING: DEVTOS FEASIBILITY DETERMINATION AND SELECTION OF AN INSTRUCTIONAL OPERATING SYSTEM

BRIEF

Requirement:

To determine the feasibility of using tactical data systems in support of individual and unit training requirements when the systems are not required for tactical operations; and to select an automated instruction (AI) system compatible with an existing Army tactical data system.

Procedure:

An existing Army tactical data system--the Developmental Tactical Operations System (DEVTOS) at Fort Hood, Texas--was analyzed to determine whether it could support computer-aided instruction (CAI). First, the specific, unique characteristics of the DEVTOS hardware and software were identified and analyzed to insure that both could support CAI. A survey and analysis of 23 existing CAI systems-languages, programs, and procedures--then determined which one would function best within DEVTOS.

Findings:

A CAI training system could be interfaced with DEVTOS without changing the system hardware configuration and without drastic reprogramming of either the AI system or the DEVTOS.

Of 23 CAI systems identified and analyzed, only one--PLANIT (Programming Language for Interacting Teaching)--met the selection criteria. From existing versions of PLANIT a single viable system was developed which interfaced with DEVTOS and provided suitable instruction programs.

Utilization of Findings:

An operational version of PLANIT has been successfully developed. Because PLANIT is portable, ARI has been able to install this author/student language program at minimal cost on three other Army data systems for different uses.

v

APPLICATION OF TACTICAL DATA SYSTEMS FOR TRAINING: DEVTOS
FEASIBILITY DETERMINATION AND SELECTION OF AN INSTRUCTIONAL
OPERATING SYSTEM

CONTENTS

AI/DEVTOS DEVELOPMENT

This report addresses the first of four interrelated tasks concerned
with determining the feasibility of applying Tactical Data Systems for
training. This task undertook a) to analyze the Developmental Tactical
Operations System DEVTOS at Fort Hood, Texas to determine if it could
support computer-aided instruction (CAI), and b) to survey and analyze
existing computer-assisted instruction programs and procedures to
determine their availability and feasibility of use within DEVTOS. This
report analyzes this DEVTOS hardware and software, describes the survey
of CAI systems, and summarizes the development and capabilities of
PLANIT--the CAI system recommended for implementation in DEVTOS.

Analysis of DEVTOS Software and Hardware

The first part of this task was to identify and analyze the unique
and specific characteristics of the DEVTOS hardware and software to insure
that the hardware and, to a lesser degree, the software of DEVTOS were
amenable to some sort of automated instruction (AI). The DEVTOS system
characteristics were next evaluated to determine ways in which DEVTOS
hardware and software would constrain the selection of an AI system.
Then the potential interface problems were determined.

The conclusion was that an AI training system (CAI) could be inter-
faced with the DEVTOS without drastic reprogramming of either the AI
system or the DEVTOS.

Survey of CAI Systems

The survey of CAI systems covered the following:

- Extensive search of the literature
- Direct communication with the authors of the systems
  being considered
- Analysis of CAI system using the DEVTOS requirements as criteria
- Analysis of CAI systems using DEVTOS-oriented capabilities as
  criteria.

As a result of the survey, 23 CAI systems were identified and ana-
lyzed. The characteristics of these systems were compared with the
requirements which had to be met by a CAI system in order for it to
operate in DEVTOS. Six of the 23 met the major requirement of being
written in either the COMPASS or FORTRAN programming language. One
system, although written in FORTRAN, was eliminated because it had no
known use at the time. The five remaining CAI systems were analyzed

against the other DEVTOS requirements and also against the minimum essential elements for a practically functional CAI system. By the time these requirements and elements had been considered, two CAI systems remained--CHIMP and PLANIT--which met the CAI requirements and elements specified. There were no known drawbacks to implementing either of these on DEVTOS. However, PLANIT--as compared to CHIMP--had a known transferability record, was currently on a Control Data Corporation (CDC) computer, and had many additional capabilities in regard to course development and recordkeeping. Consequently, PLANIT was logically derived as the CAI system recommended for implementation in DEVTOS. The U.S. Army Research Institute (ARI) proceeded to obtain several versions of PLANIT. With these as a point of departure ARI developed a single viable system which would interface with the available hardware and software within the constraints of the Fort Hood system, produce instructional programs with the requirements dictated by the courseware strategies, and utilize and adapt the best segments of the different versions of PLANIT.

Instructional System Development and Basic PLANIT Capabilities

As part of this project, a survey of existing AI systems was conducted to determine which, if any, would meet the needs of the project. Project needs included (a) an AI system integrated into a tactical data system (DEVTOS) which was not specifically designed for AI, and (b) courseware developed which the AI software selected for the experiment could execute and modify. This section discusses the integration of this system within the operational environment at the test facility, Fort Hood, Texas.

As one of the first integration activities, a proposal was reviewed which contained a plan for integration of an AI system within DEVTOS in which the current operating system would remain basically intact and an AI system would be customized to operate within that system.[1] For several reasons this was not felt to be the best approach. First a customized AI system would be untested and would probably require extensive checkout. Secondly, a customized AI system would not be the off-the-shelf product desired, and its transferability and utility within other tactical systems (assuming project results indicated that AI in tactical computers was feasible) would be severely curtailed. Availability of the AI capabilities and functions considered necessary to meet project commitments was questionable. Finally, upon completion of Task 1, it was concluded that an existing AI system PLANIT could be integrated into DEVTOS to provide the functional capabilities for courseware development, modification, instruction presentation, and student recordkeeping--all considered necessary in an AI system. (Student recordkeeping later proved to be a serious deficiency in all versions of PLANIT.)

---

[1]From Bunker-Ramo Technical Note "MASSTER Test 122--Computer Assisted Instruction (CAI) Concept Paper," February 1973, prepared for the U. S. Army Computer Systems Command.

The AI system selected was PLANIT (Programming LANguage for Inter-
active Teaching), considered to be the most effective and least costly
AI system with which to meet project commitments. PLANIT, as a computer
program system for AI, provides an AI author language and computer pro-
grams that make a variety of operations available to the user. Briefly,
PLANIT allows an author to enter instructional material into the computer--
interactively on-line, or off-line as card inputs--and to store this
material in designated sequences on appropriate storage devices. The
author may enter his material in any of several formats and can imme-
diately review, edit, and revise the instructional material as necessary.
Completed instructional programs (AI lessons) are presented to a student
at the remote terminal through the execution facility of the system.

PLANIT contains an interactive calculation language. The complete
computation language may be used by an author interactively or may be
used as instructions to PLANIT in an instructional program. A subset of
the calculation language is available to the student as he executes
instructional material, unless its use is prohibited by the lesson author.
The student can construct his own mathematical functions as well as use
the functions an author makes available to him.

PLANIT also permits an author to specify alternative actions to be
executed based on student response. The criteria for conditional feed-
back or branching can be based upon the cumulative performance records
of the student which are automatically kept by PLANIT or upon records
kept by programming statements in the lesson. PLANIT also provides
response-processing routines to aid in matching student responses against
the author's anticipated responses. These aids to response-matching
include phonetic equivalency comparisons, equating uppercase and lowercase
characters, searching for key words in the response, searching for key
characters in the response, automatic matching of numeric equivalents,
and automatic matching of algebraically equivalent expressions.

Finally, PLANIT provides for on-line interactive control of off-line
utility operations. This includes generating instructional material from
or onto cards, obtaining a listing of completed material, and performing
a variety of maintenance tasks related to student performance records
and/or manipulation of instructional material stored on disk or tape.

The System Development Corporation (SDC) supplied a computer tape
of its 1970 version of PLANIT to ARI in February 1973 to permit ARI
personnel to become familiar with PLANIT coding. During March-June 1973
SDC served primarily as a consultant toward the PLANIT/DEVTOS integra-
tion, reviewing and making recommendations concerning a second
functional paper in which a modified PLANIT would be integrated into the
existing DEVTOS. SDC pointed out that because of PLANIT's internal
logical design, whereby an interdependency of operations exists among
functional areas, it was in the best interests of the project to partition
and overlay an existing version of PLANIT within DEVTOS rather than
attempt to modify or delete existing functional capabilities.

ARI investigated the possibility of letting a contract with a software house for developing a fully operational PLANIT with Machine Interface I/O Program (MIOPS). Cost estimates were too high and estimated completion date too distant to make this approach feasible, particularly since an anticipated early release of the National Science Foundation (NSF)/Purdue University version of PLANIT in early June 1973 never materialized. Thus, it became necessary for ARI to develop a workable PLANIT. The writing of the MIOPS and the integration of these PLANIT components with the communication package was contracted out to the Bunker-Ramo Corporation (BRC) under the aegis of the U. S. Army Computer Systems Command (USACSC) in accordance with a statement of work supplied by ARI. This software interface permitted the use of the (CDC) general operating system MSOS to drive the DEVTOS equipment, thereby eliminating the need to rely on the special DEVTOS software system.

The early possession of a version of PLANIT permitted ARI to gain some familiarity with PLANIT logic, but this early version had known deficiencies, including an inability to properly maintain student records, which prevented its further programming development. Although the NSF/Purdue version (later called PLANIT 1.0) was not available, ARI obtained a copy of the Michigan State University (MSU) version--a lineal descendent of the Freiburg version, as is the NSF/Purdue version. With some consultation with MSU personnel (particularly Dr. Rahimi), ARI installed an improved version of the MSU PLANIT modified to run in 20K of memory, first at the ARI facility in Arlington, Virginia and later on the DEVTOS at Fort Hood. The final ARI version of PLANIT was designated PLANIT 1.1 by the PLANIT Users Executive Committee during the fall of 1974.

During July-August 1973, specific functional problems were identified during field tests and integration of AI software and AI courseware. Inhouse tests at ARI, joint SDC-ARI tests at ARI, and joint tests at Fort Hood determined that a number of PLANIT software functions were not working according to design intent, i.e., the earlier SDC documented concepts of PLANIT and the current available PLANIT versions differed in several major aspects. These determinations were based on on-line runs using AI courseware and/or special test cases (PLANIT frame sequences) constructed by the lesson authors. Design intent was determined by functional specifications presented in the PLANIT Author's Guide (SDC TM(L)-4422/001/01) [2] and the PLANIT Language Reference Manual (SDC TM(L)-4422/002/01. [3] These were used as the standard against which

---

[2]Bennik, F. D., and Frye, C. H. PLANIT author's guide (TM(L)-4422/001/01). Santa Monica, Calif.: System Development Corporation, October 1, 1970.

[3]Butler, A. K., and Frye, C. H. PLANIT language reference manual (TM(L)-4422/002/01). Santa Monica, Calif.: System Development Corporation, October 1, 1970.

to measure PLANIT's functional performance because: (1) they define system reaction under given conditions of courseware encoding; (2) an interim version of these documents, co-authored by the PLANIT designer and his staff, was used as the PLANIT functional specification at the time SDC undertook to make PLANIT a machine-transferable system under contract to the National Science Foundation; and (3) AI project lession authors needed a language specification for designing and encoding AI materials. Im ambiguous situations, the functional capabilities and design intent were determined by telephone contact with the designer of NSF PLANIT, Dr. Charles H. Frye, PLANIT Director, Northwest Regional Educational Laboratory, Portland, Oregon.

Table 1 summarizes the functional problems identified and alleviated by joint effort before the AI field test began. The problems are categorized functionally, according to whether they were problems of presentation control, response acceptance and processing, program control, decision functions, or system suppor'. For an explanation of the PLANIT language shown in parentheses in the table, refer to the PLANIT documentation referenced above.

Of the problems listed in Table 1, those dealing with decision frames were solved primarily by installing a rewritten PLANIT decision logic into the Fort Hood PLANIT. This deck was rewritten at Northwest Regional Educational Laboratory and was installed by ARI at Fort Hood. Another change solved the problem with multiple-decision statements of pattern form.

Another problem for users in the student mode was slow response time, ranging from 6 to 12 seconds for response frames, and 6 to 14 seconds--up to as much as 50 seconds--for decision frames. These response times were speeded up to tolerable levels for the AI field experiment through ARI's use of the high speed drum for handling the overlays.

It should not be construed that all functional capabilities of PLANIT were subjected to a detailed verification, as this was neither the case nor the intent of this phase of the project activity. The results of this integration effort did, however, produce a viable PLANIT that fully supported the requirements of this project. This is the version of PLANIT subsequently designated Version 1.1 by the PLANIT Users Group. The majority of presentation, response processing, program control, and decision functions of PLANIT Version 1.1 appear to operate correctly. However, a number of PLANIT calculation capabilities (e.g., matrix, algebraic, and review statements) were not tested in this experiment. Further test of system utility functions (off-line support activities) is also warranted.

Table 1

PLANIT DEVELOPMENT MILESTONES DURING AI FIELD TEST

PROBLEM AREA AND PROBLEM	DISPOSITION
Presentation	
PLANIT line skip control character was not working at student consoles. The effect of this was to undo the display formats created during lesson building. This occurred only at Fort Hood.	CRT driver was modified by ARI to correct this condition at Fort Hood.
PLANIT operator's message (BUSY) appeared intermittently during lesson execution.	Result of ARI changes to PLANIT; corrected at Fort Hood.
Response Acceptance and Processing	
Use of response processing function (TEXT) in lessons caused abort of PLANIT.	TEXT function stated to be operating in Freiberg version of PLANIT did not work. TEXT function removed from all courseware.
No match of a correct numeric response where the first character was a decimal point (e.g., the student's response .0025 would not match the author's A+ .0025).	Modified courseware to space between + and decimal point (e.g., A+ .0025)
Response to multiple-choice frame other than a letter tag from multiple-choice list caused PLANIT to abort, rather than printing the prompt CHOOSE ONE OF THE ABOVE LETTERS.	Fixed at ARI prior to Fort Hood by adding additional card in cold start deck.
Program Control Functions	
Lesson control word (RELATED) operated inconsistently in constructed-response and multiple-choice frames.	RELATED found to operate correctly in Group 3 of Q frames and Group 4 of M frames.
Timed pacing (WAIT) found inoperative.	Worked when used with CRT's after removing an MSU correction.
Logic for automatic feedback of correct answer found to be reversed--correct answers prefaced by a number tag were treated as literals for feedback, whereas correct answers prefaced by a letter tag were treated as a numeric expression in feedback.	Fixed at ARI prior to Fort Hood.
Second or subsequent unanticipated response tags caused PLANIT to abort.	Fixed at ARI prior to Fort Hood.
Decision Frames' Logic and Records	
Logic of summary form of decision statement (RIGHT, WRONG, SEEN) found to be incorrect or inconsistent in operation.	Tested, fixed by Dr. Frye, and retested at Fort Hood.
Logic of pattern form found incorrect when operating over a series of decision statements.	Tested, fixed by Dr. Frye, and retested at Fort Hood.
Logic of compound decision statements with clauses connected by AND and OR treated both as OR clauses.	Tested, fixed by Dr. Frye, and retested at Fort Hood.
Frame size was too small.	Increase of frame size at Fort Hood.

Table I continued

PROBLEM AREA AND PROBLEM	DISPOSITION
System Support	
Extraneous characters were intermittently inserted into lesson disk file when lesson was executed by an author. If characters had overwritten frame control characters, subsequent lesson execution would abort PLANIT. If characters had overwritten display and feedback messages, subsequent lesson execution displayed the extraneous characters.	Problem identified at Fort Hood.
Intermittent problems occurred after deleting a lesson and its student records from a disk, wherein subsequent lesson construction left both named _and_ un-named lesson files on disk.	Procedure changes at Fort Hood eliminated the problem.
Use of command to name a lesson (SAVE) frequently resulted in unnamed lessons left on disk with further access denied.	Procedure changes at Fort Hood eliminated the problem.
Each successive punch out of a given lesson shifted lines one space to the right, until display width was exceeded.	Fixed at Fort Hood.
Failed to store lessons onto tape or to retrieve lessons from tape.	Alternative procedure was installed at Fort Hood.
Failed to transfer student performance records onto tape, and to list student records on high-speed printer.	Alternative procedure was installed at Fort Hood.
Interfaced unreliably with communications software and system hardware.	Fixed at Fort Hood by development of new CRT software prior to AI experiment.
PLANIT intermittently dropped lesson frames for unknown reasons during online editing and building.	Status uncertain--precise reason for problem uncertain; problem alleviated by updating card decks and rebuilding lesson.

This detailed section describes the findings from an examination of the hardware and software characteristics of DEVTOS, identifies the requirements which had to be met by an automated instruction (AI) program to operate within the DEVTOS system, describes some potential interface problems, and discusses some of the characteristics of the DEVTOS hardware and software.

The general operation was as follows: Students used cathode ray tube (CRT) terminals for both input and output. Textual information, questions, and remedial feedback were displayed to the student on his CRT. Using the typewriter-like keyboard on the CRT, the student typed in his answers to questions or used predetermined codes to start or advance through the lesson. The computer responded appropriately with new displays--new pages of text, feedback, or remedial data. Ten CRT terminals were used for students with two more used for monitoring and controlling purposes. Student results could be output on hard copy immediately after a course was completed, using the typewriter which was paired with each CRT.

Evaluation of DEVTOS Hardware and Software for AI Use

Certain general criteria must be met by any computer system for the system to support AI. The DEVTOS hardware and software were evaluated in terms of these criteria to determine if AI was possible on the DEVTOS configuration. The broad characteristics required for AI are identified in the following blocks and the DEVTOS CDC 3300 capability is discussed relative to these characteristics.

1. <u>A means of inputting AI system and AI course information into the computer must exist</u>. As the CDC 3300 computer of the DEVTOS has both card reader and magnetic tape inputs, input was no problem.

2. <u>Adequate storage space (memory) to store the AI system and AI course materials is required</u>. The DEVTOS computer system has a total storage capacity in excess of 5 million words or 20 million characters. Assuming 1000 characters per course frame, there is storage for 20,000 frames. This would probably accommodate over 200 hours of instruction, ample storage for the course materials. Most AI software systems require about 200,000 characters of space for operation, which is well within the DEVTOS capacity.

3. <u>An adequate terminal device must be available for the student to use</u>. DEVTCS has teletype, typewriter, and CRT terminal inputs. Any of these are adequate. The best of these is the CRT terminal because of its speed, silent operation, and availability. Also, the CRT, unlike the typewriter, prevents the student from looking back for answers to criterion items. Since he only has the current display frame available, the student is forced to learn the material.

4. <u>A means must exist to pass control to the AI system so it can</u>
operate. DEVTOS was designed in a modular fashion to facilitate changing
and adding features. It has a complete operating system and executive
incorporated into it. While all the existing software is custom built,
the modular design and table driven concept make it possible to add a
specific applications package such as an AI system. The task of inter-
facing the DEVTOS software and the AI system software could have been a
significant problem if the AI system were poorly documented, did not
identify the interface points, or were not designed to be machine trans-
ferable; therefore documentation and machine transferability were
included among the requirements which the candidate public domain CAI
systems were asked to meet. An AI system with good documentation, machine
transferability, and interface points would operate within DEVTOS under
the control of the DEVTOS executive. The Tactical System Development
Group (TSDG) at Fort Hood, which programs and maintains DEVTOS, cooper-
ated closely to insure this coordination.

Requirements for an AI System

After determining that DEVTOS met the four criteria for supporting
AI, the DEVTOS system was examined to determine what constraints its
hardware and software would place on an AI system. These constraints
then became the AI requirements:

1. The AI system must be able to operate on the CDC 3300 computer;
either by finding an AI system already operating on a CDC 3000 series
computer or by installing a machine transferrable AI system on the 3300.

2. The AI system must be written in either the COMPASS or FORTRAN
programming language since these are the only languages usable with
DEVTOS.

3. The AI system must operate within the DEVTOS hardware limitations:

   24-bit words with 4 @ 6-bit bytes per word
   81,920 words of core memory
   4,157,440 words of auxiliary disk storage
   1,048,576 words of auxiliary drum storage
   5 magnetic tape units
   1 card reader
   1 line printer

See the section on DEVTOS hardware characteristics for further details
of the hardware limitations, characteristics, and a discussion on their
impact on AI.

4. The AI system must be operable or modifiable to operate under the
DEVTOS executive program. This includes such things as having defined
entrant and reentrant points and having the ability to be segmented into
4K modules.

5. The AI system must be able to handle at least 10 students on-line at CRT terminals. In addition, a terminal should be available for control purposes and another for monitor purposes.

6. The AI system must have adequate documentation so that programming interfaces may be identified, specified, programmed, and tested

7. The AI system must be able to process or use the following characters which are available on the CRT:

a)	A-Z	Letters	n)	;	Semicolon
b)	0-9	Numbers	o)	/	Slash
c)		Space	p)	:	Colon
d)	.	Period	q)	@	At
e)	,	Comma	r)	!	Exclamation
f)	-	Hyphen (or minus)	s)	?	Question
g)	+	Plus	t)	"	Quotes
h)	=	Equals	u)	#	Number
i)	%	Percent	v)	<	Less than
j)	*	Asterisk	w)	>	Greater than
k)	$	Dollar	x)	'	Prime
l)	(	Left parenthesis	y)	&	And
m)	)	Right parenthesis			

In DEVTOS a) through k) were the only legal characters for use in message texts. This would not preclude the use of the full set in AI display frames but did limit the textual responses from the students to those characters in a) through k).

8. The AI system must be capable of producing on-line student results at a hard copy terminal, to satisfy the research design requirement that hard-copy feedback be available immediately upon course completion.

Apparent Interface Problems

Even before the actual CAI system had been selected, certain interface problems appeared likely on the basis of available data. The underlying assumption was that the AI system would be an entity complete in itself which would perform all the AI functions.

The first problem was with the student interface. Since the preferred student interface was with the CRT display terminal, several programming modifications were necessary. The Remote Station Data Terminal (RSDT) had to be reprogrammed to permit both input and output from the CRT. The formats used for the AI display also had to be programmed into the 1700. The CDC 3300 required reprogramming in both the input and output interfaces with the 1700. If the inputs were to be error and validity checked, this too would have to be programmed. (If an AI system were to have its own input error-checking capability, no additional error-checking programming would be required.)

Interface between the DEVTOS executive and the AI package was another programming task which had to be accomplished. The DEVTOS executive has certain conventions in passing control to an application program which must be met, i.e., the driving table. The interrupt processing functions of the DEVTOS executive are based on tactical assumptions which are not always valid for AI training purposes. Other interface considerations, such as character incompatabilities, required that small translator routines be written so that the DEVTOS executive and the AI package could accurately communicate with each other.

The interface between the AI package and the data base of frame information is typically a machine-specific program which is tailored when the AI package is installed. The AI package required a translator routine to enable it to retrieve frames of textual material.

The interface between the AI package and the input and output devices was another piece of required machine-specific software. This consisted of a program linkage to enable the AI package to talk to and understand the terminal devices, probably in the nature of a simple translator routine.

A processor needed to be programmed to read the AI system into the computer and to read in the instructional frames. These would have to be read from either cards or tape and stored on disk or drum. The storage locations of the AI system and the data base of instructional material would have to be known by the DEVTOS executive and the AI system.

The AI system had to be divided into 4096-word segments in order to meet the DEVTOS dynamic core allocation requirements and to switch information from auxiliary storage to main storage. BRC suggested that this be done manually since no computer utility program existed to do it automatically.

A special copy of the DEVTOS software had to be created to incorporate appropriate programming for the above considerations. In addition, the tactical functions had to be blocked to keep them from being turned on accidentally by a student during AI training sessions.

DEVTOS Hardware Characteristics

The DEVTOS is a highly mobile system. All hardware components of the system are installed in vans or trailers.

Central Computing Center (CCC)

The CCC is the name given to the central computer complex and the vans which house the computers. The computer is basically a CDC 3300 which is a large-scale business-oriented machine, with the floating point arithmetic, multiprogramming, and real-time communications options installed. The computer has the following characteristics:

- 11 -

Word size:	24 bits
Number of characters/word:	4
Number of bits/per byte (character):	6
Character coding:	BCD
Memory capacity (words):	81,920
Memory capacity (characters):	327,680

The memory capacity had been expanded to a total of 5,287,936 words (21,151,744 characters) by the addition of both disc and drum auxiliary memory devices. Two CDC 854 disk drives have the combined capacity of 4,157,440 words or 16,629,760 characters. (Each disc can hold 2,078,720 words or 8,314,880 characters.) The average access time for the disk is 107.5 milliseconds. The drum is a CDC 863 mass storage drum which can contain 1,048,576 words (4,194,304 characters). The average access time for the drum is 17 milliseconds.

The computer has the following peripheral devices:

1 Operator Console
1 Card Reader capable of reading 1200 cards per minute
1 Card Punch capable of punching 250 cards per minute
5 Magnetic Tape Drives (7 track) which are IBM compatible
1 Line Printer capable of printing 1000 lines a minute
Plus appropriate controllers for the devices

The principal external interface of the central computer is by means of data set adapters which couple with the crypto equipment. The crypto equipment is linked by either radio or telephone lines to crypto equipment at the Remote Station Data Terminal (RSDT). The RSDT contains a CDC 1700 computer which served as a terminal device controller, message buffer and message switching device as well as a repository for various message format skeletons. Figure 1 shows the relationship between the Central Computer, Remote Computer and terminal User Input Output Devices (UIOD's).

The cryptographic link suggested that in the AI application all AI information would pass through the encryption/decryption equipment (see Figure 1). This had to be explored to determine if (1) all data would pass through the crypto equipment, (2) the mass of AI data would have an adverse impact on the malfunction rate of that equipment and (3) if some device or technique could be used to bypass the crypto equipment.

Remote Station Data Terminal (RSDT)

The RSDT contains the CDC 1700 computer, its associated crypto equipment, the terminal devices (UIOD's) and the necessary cables and controllers to operate the UIOD's. The 1700 computer is a small commercially available computer frequently used for process control in manufacturing applications and for terminal control and message switching as it is in DEVTOS. The 1700 has the following characteristics:

12

Figure 1. Information flow through DEVTOS

Within the figure:

Central Computer Complex

CCC

Disks & Drum

Data Base & Pro- grams

Card Reader

Card Punch

Line Printer

CENTRAL COMPUTER

(81,920 24-bit words of Core Memory, plus Auxiliary Disc, Drum and Magnetic Tape Storage)

Input/ Output Buffers

General and Special Processes

Executive Program

Operator Console

Tape Pool

Crypto Equip- ment

Crypto Equip- ment

RSDT

REMOTE COMPUTER

(24,576 16-bit words of Core Memory)

Input/ Output Buffers

Message Formats

Message- Switching Logic

UIOD

Display Station (CRT)

Type- writer

Operator

Hard Copy Output

Adapted from BRC Tactical Operations System Software pamphlet.

- 13 -

Number of bits per character	8
Type of character coding	ASCII
Number of characters per word	2
Number of bits per word	16
Number of words per module	4096 (8192 characters)
Number of storage modules	6
Total number of words available	24,576 (49,152 characters)

CRT Display Unit. The Display Unit consists of a CRT screen and a keyboard (Figure 2). Information typed at the keyboard is displayed on the screen as each key is depressed. Information typed on the screen using the keyboard can then be transmitted from the display unit by depressing the "SEND" key. Information transmitted to the display unit by the system is displayed on the screen; thus, the display unit can transmit data and receive stored format-display information.

Screen. The screen is similar to a small television screen. Information typed at the display unit keyboard is displayed in a rectangle consisting of 20 lines of 50 characters each, for a total display capability of 1000 characters.

Operator Communication Field (OCF). The first four character positions on the screen (positions 1, 2, 3, and 4 of line 1) are reserved for operator communication codes transmitted by and to the operator during message receipt and transmission. These four character positions are referred to as the Operator Communication Field (OCF).

Cursor. A cursor appears on the screen as a one-character underline. The purpose of the cursor is to inform the operator where on the screen the next character will appear when a key is depressed. For example, in typing the word "RETRIEVE", the characters typed and the cursor appear as follows:

RETRIEV_

The next letter to be typed is "E" and will appear directly above the cursor when that key is depressed. At each operation of a key or space bar, the position of the cursor advances one space. When the cursor reaches the end of a line, it automatically moves to the first position of the next line to accomplish a carriage return. When the cursor reaches the end of the last line on the CRT screen it automatically returns to the first position of the first line. If the operator makes a typing error, he repositions the cursor under the incorrect character and either blanks out the error or types in the correct character. The correct character, or a blank, takes the place of the erroneous character.
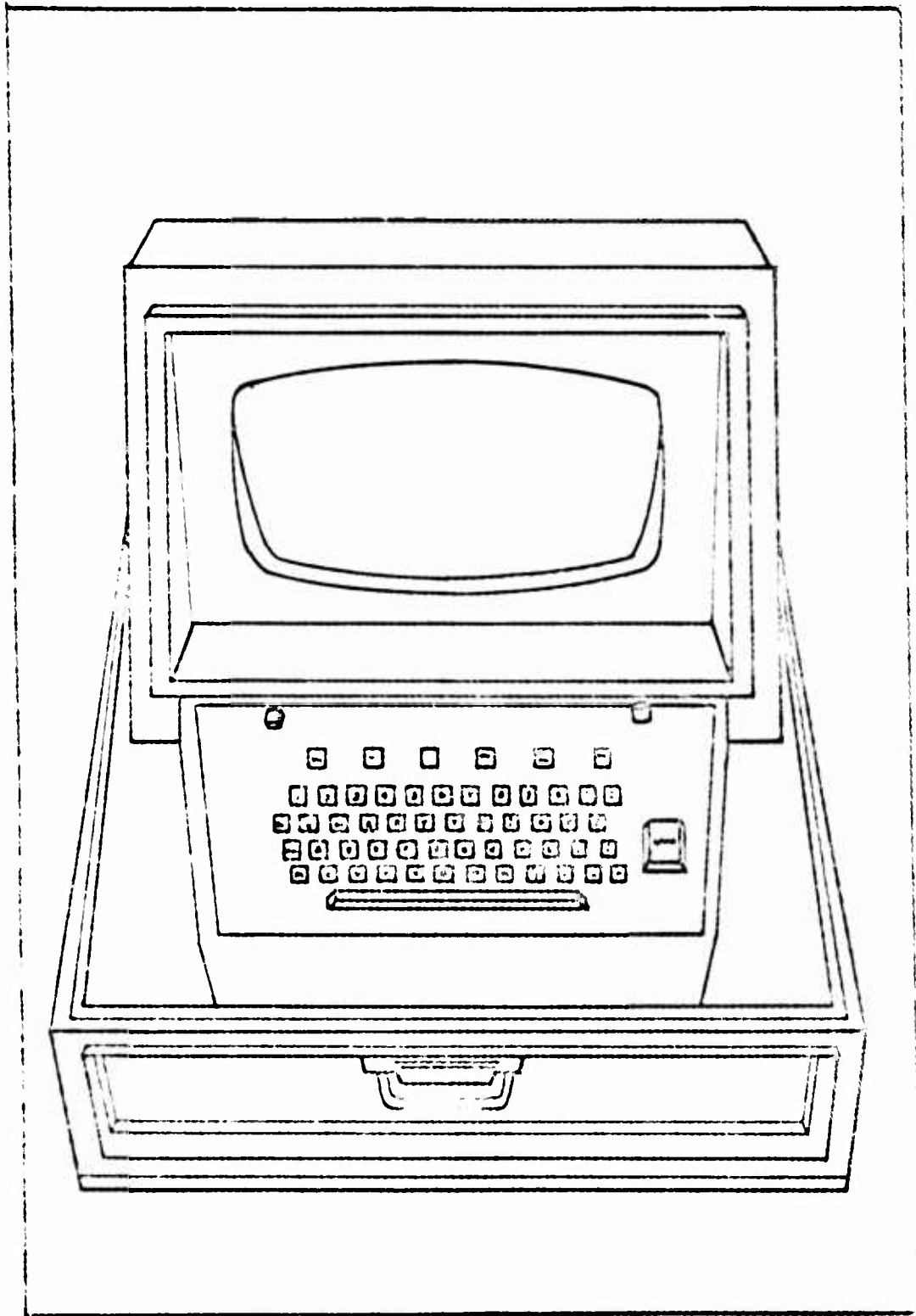
Figure 2.   Display unit

Display Unit Keyboard. Except for insignificant differences in control keys and the absence of lower case (small) letters, the Display Unit keyboard (Figure 5) is similar to a standard typewriter keyboard. Control keys on the keyboard perform as follows:

(1) Clear. Depressing of the CLEAR key clears all data from the CRT screen. The cursor is moved to the upper left corner of the screen.

(2) Backspace. Depression of the BKSP key moves the cursor one space back without changing displayed data.

(3) Reset. Depression of the RESET key moves the cursor to the upper left corner without changing displayed data.

(4) Return. Depression of the RETURN key advances the cursor to the beginning of the next line without changing displayed data. This key performs the same function as a manual carriage return on a typewriter.

(5) Depression of either SHIFT key enables entry of the upper symbol on the 2-symbol keys. Operation of the single-symbol key is not affected by the SHIFT keys; all alphabetic symbols (letters) are displayed in upper case (capital) form. The SHIFT keys do not lock.

(6) Space Bar. Depression of the SPACE BAR moves the cursor one space forward without changing displayed data. This key has the same function as the space bar on a typewriter.

(7) Repeat. Depression of the REPT key causes the action of another depressed key to be repeated at a rate of eight per second while the REPEAT key is depressed. Keys not affected by the REPEAT key are CLEAR, SHIFT, RESET and SEND.

(8) Erase. The operation of the ERASE key erases any character at the position of the cursor and advances the cursor one space.

(9) On/Off Intensity. The ON/OFF INTENSITY knob is located at the right side of the Display Unit. Rotating the knob clockwise turns the Display Unit screen on. Further rotation increases the intensity of the displayed symbols. When the ON/OFF INTENSITY control is off data can still be entered at the keyboard and transmitted.

(10) Send. Depression of the SEND key writes an end-of-message symbol (Δ) at the cursor position, and caused information displayed on the screen to be transmitted. Prior to depressing the SEND key, the operator must ensure that the cursor is in position 4 of the Operator Communication Field (character positions 1, 2, 3, and 4 of line 1 of the CRT screen). The operator should not hold the SEND key down since this degrades the operation of the RSDT.
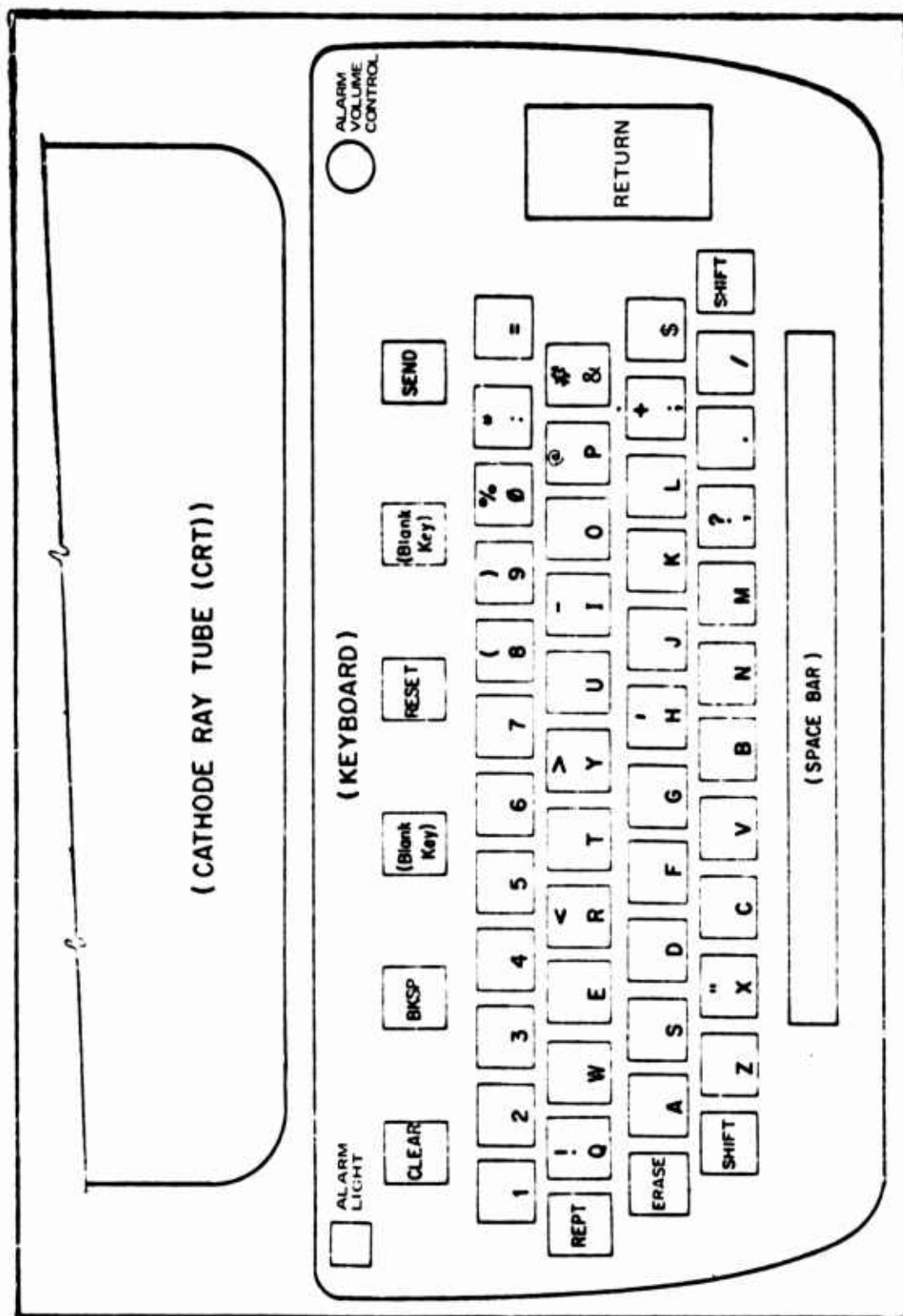
Figure 3. Display unit keyboard

Typewriter-Printer. The Typewriter-Printer (Figure 4) is a modified office typewriter. The main function of the typewriter is to print output messages, which it does at the rate of 15 characters per second. Information typed on the typewriter is transmitted to the system; however, it can transmit only short codes such as "ACK," "RPT" or "HLT."

Keyboard. The Typewriter-Printer prints upper case alphabetic characters. It is not necessary to depress the shift key to type response codes.

Typewriter control settings:

1) Line space lever. This lever controls the line space movement of the platen and has two settings: double space and single space. This lever is to be set at single space.

(2) Left margin stop. The left margin stop is to be set at margin scale position five.

(3) Right margin stop. The right margin stop is to be set at the right end of the margin scale.

(4) Multiple copy control lever. To compensate for additional copies, this lever is moved from the forward position toward the rear. This lever is to be set at the second position for 2- to 5- part paper.

(5) Intensity selector. The intensity selector controls the force with which the typing element strikes the paper. The intensity selector is to be set at position 3 for 2- to 5- part paper.

(6) On/Off key. Prior to operation the operator must insure that this key is in the ON position. During operation the operator will insure that this key remains in the ON position. <u>Placing this key in the OFF position during the operation can result in the loss of data</u>.

DEVTOS Software Characteristics

The software for DEVTOS is almost entirely custom software which was designed, programmed, implemented and maintained by a highly qualified team of Bunker Ramo Corporation (BRC) software specialists. The 3300 software is written in COMPASS and FORTRAN while the 1700 programs are written in machine language. For optimum efficiency and speed of execution, BRC prefers to use COMPASS in most of the applications.

Software system. The following excerpt for BRC's TOS software brochure describes the software philosophy utilized in DEVTOS.

DEVTOS Software Organization. The DEVTOS is a large-scale (240,000 computer instructions) command and control information system. The DEVTOS software system consists essentially of functional capabilities, to process the special tasks which fulfill commanders' information needs during tactical operations and operating system programs, to control system and data manipulation processing and to perform the processing common to the tactical information (functional) areas.

Two system concepts in the DEVTOS give flexibility in program operation and modification -- the "modular" operation structure, and the "table-driven" processing techniques. In the modular approach, individual processing tasks are incorporated as self-contained, procedurally independent segments or "modules." Modules similar or related in performance are grouped as "functions" for efficient system operation. A module may be added, modified, or deleted with minimal effect on the overall system structure or its operation.

Table-driven processing imparts desirable flexibility to a developing system. Processing paths dependent on results of validity tests, criteria grouping, data storage in multiple files, etc., are selected in tests and instructions contained in "driving tables." Changes in tactical information requirements can be incorporated without restructure of the basic system or additional programming, by adding, modifying, or deleting driving tables or table parameters.

Operating System--The Supervisor Software. The Operating System consists of the Executive, General Processes, and Off-line Support programs.

The Executive Program, as the on-line software "director" for the DEVTOS, dynamically controls all system processing -- including priority scheduling, simultaneous processing of several "transactions" and the allocation of computer memory and mass storage; manages all equipment resources, and maintains real-time communication simultaneously on multiple channels; maintains the data base from instructions obtained from General or Special Processes; and maintains a log of system operations, and collects operating statistics for performance analysis.

The General Processes provide services common to messages in all of the tactical information areas (i.e., the functional areas). The direction of this processing is prescribed by the driving tables. The programs of the General Processes perform such tasks as to validate all messages, with error notification to originators; maintains files, retrieve and compare data; format and control dissemination of data in response to specific inquiries; automatically format and control dissemination of data in response to Standing Requests for Information (SRI); and check security and restrictions for all transactions and all users sending or receiving information.

The Off-Line Support programs perform off-line tasks, essential to the on-line processing of the DEVTOS, such as to translate into internal computer language the external language used by analysis in preparing the driving tables; reduce data and generate statistical reports; and provide utility services, e.g., system tape preparation.

LEFT-RIGHT MARGIN

ON/OFF KEY

PAPER RELEASE

LINE SPACE

MULTICOPY CONTROL

INTENSITY
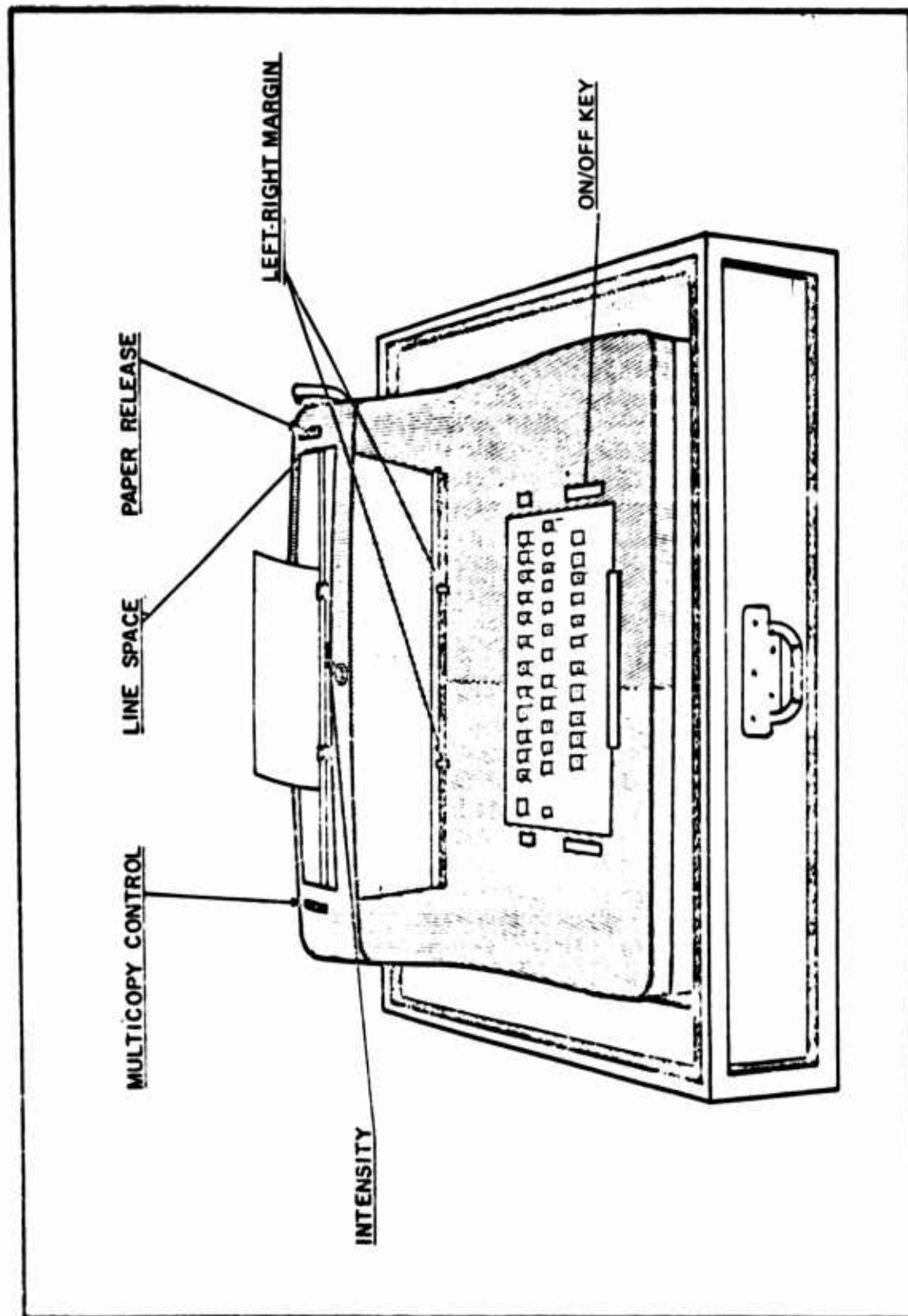
Figure 4. Typewriter-printer

The organization of the DEVTOS software system in Figure 5 shows the program areas and program sizes as of September 1972. For a detailed description of each program area, the reader is referred to the Tactical Operation System (TOS) Software System Description,[4] a copy of which is in the AI Project Library.

Additional Comments. Generally, all on-line programs must be divided into 4K modules in order for the dynamic allocation of core and the swapping functions to operate.

The FORTRAN compiler generates about 5 computer words of code for each FORTRAN statement. In other words, a FORTRAN program of 1000 statements would generate about 5000 words of machine language code and consequently would not fit into a 4096 word module.

The CCC is programmed to log selectable combinations of terminal transactions on a magnetic tape. The subsequent processing of the log tape enables analysis of such things as amount of time the terminals were in use, amount of time they were waiting on operator input, length of time it took for each operator to make his inputs, number of input errors, etc. The DEVTOS is quite well instrumented; however, the people there advise that attempts to log all transactions slow down the response time.

The DEVTOS executive program and operating system are unique to DEVTOS. It was reported that there is no documentation available on it other than design specifications which are not necessarily accurate or up to date. BRC has the only real knowledge and expertise in this area.

Some of the restrictions in the DEVTOS are that the characters "/" and ";" are field delimiters and not available for use (without reprogramming). The computer has an idiosyncrasy with the colon. The internal code for a colon is 12 octal, the tape code for a zero is 12 octal, thus when colons are dumped on tape and subsequently read, they become zeros.

Synopsis of DEVTOS Operational Philosophy

The DEVTOS system is designed to permit on-line storage, update, and retrieval of specific tactical and intelligence information. The philosophy has been to automate the existing manually reported information.

The storage and update process functions are as follows:

Specific report formats are defined and stored in the 1700 computer. A UIOD operator desiring to input a certain type

---

[4] Tactical Operations System (TOS) Software System Description. Major Curtis S. Tomlin, USACSC-TSDC, Systems Analysis Branch, Fort Hood, Texas, January 1972.

Figure 5. DEVTOS software system organization
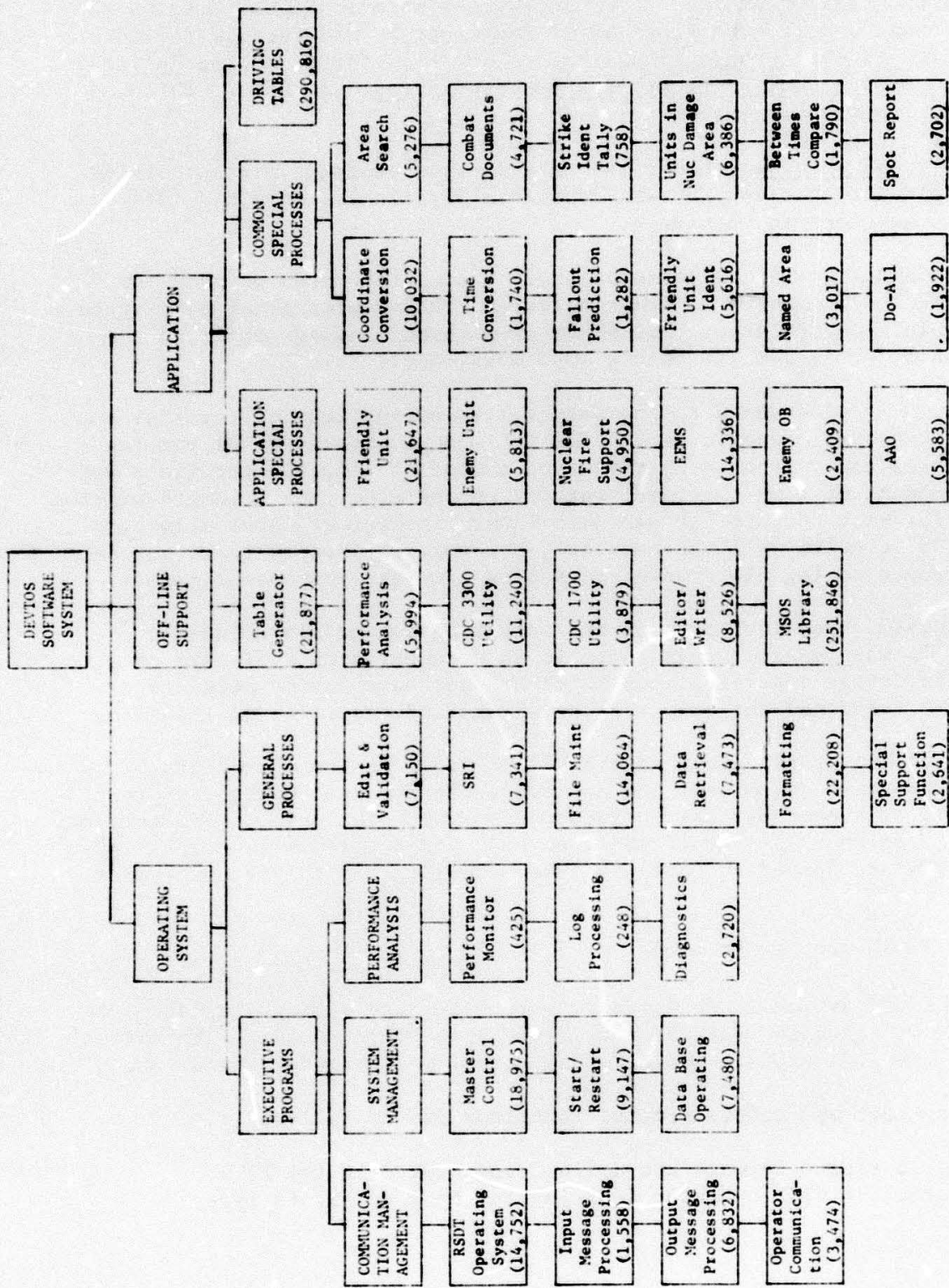
Total number of computer instructions = 820,625

as of September 1972

- 22 -

of data requests the appropriate format by typing in the appropriate code. The 1700 computer analyzes the code and responds with the appropriate message or report format which is displayed on the CRT screen. The operator then spaces the cursor down to the appropriate space in the format and types in the information he wants in the data base. The space (or field) in the format is marked by the field delimiters "/" and ";". The slash indicates the beginning of the field and the semicolon designates the end. The operator must type his information within those delimiters.

When the operator has completely filled in the format or filled in what is appropriate, he adds the message precedence, security classification, and hard copy indicator and "sends" the message to the 3300 through the 1700. Upon receipt of the message, the 3300 sends back an "ACK" to acknowledge receipt of the message. After error checking the message, the 3300 sends back a "COR" or an "ERR" depending on whether the message was correct or in error. If the message is correct, it is further processed and stored in the data base. If the message is incorrect, up to five error messages will be listed on the last 5 lines of the CRT display screen.

The process used by the operator to retrieve data is similar. He types in a code which the 1700 interprets and responds with a message format. The format will be a query format into which the operator will insert certain parameters. When the message is sent, the 3300 will error check and process the query. However, the response to the query will come back to the typewriter for a hard copy of the information. The output will be in a specific predetermined format. The data input and query processes are displayed in Figure 1 on page 13.

Other retrievals possible are called Special Process Request Messages (SPR) and Standing Request for Information (SRI). The SPR permits combining functions of several formats or requesting special calculations. The SRI permits recurring reports to be generated on a specific time table. There is also a capability to relay messages from one station to another. The processing of these messages is similar to the query processing.

An interesting feature of the information transfer from CRT to 3300 is the function performed by the 1700 computer. The 1700 has all the format skeletons stored in its memory. When a format is called for, the 1700 provides it. When the format is filled in and sent from the CRT, the 1700 strips out the data, discarding the format skeleton, adds a key to tell the 3300 which format was used, and sends the key and the data to the 3300. The 3300 checks the key, knows which format was used and proceeds to edit and error check the data accordingly. This saves the repetitive transmission of meaningless formats and shortens the transmissions considerably. Also, if a typewriter or CRT fails, the RSDT is programmed to make that device unavailable and notify a designated CRT or typewriter. I/O devices may be turned back on by notifying the RSDT via one of the operational typewriters or CRTs that the particular device is again available.

SURVEY OF CAI SYSTEMS

This section describes in detail the findings from the survey and analysis of existing computer-assisted instruction (CAI) systems for availability and feasibility of implementation within the Developmental Tactical Operations System (DEVTOS). This survey and analysis provided the basis for selections by the U. S. Army Research Institute and the U. S. Army Computer System Command Tactical Systems Development Group (USACSC TSDG) of the CAI system used on the DEVTOS system. Parts of this section were written before the final debugging of Michigan State University and NSF versions of PLANIT and are presented to give a feeling for the process involved.

Identifying CAI Systems

A survey which included the University of California at Los Angeles (UCLA) and System Development Corporation (SDC) libraries covered journals, indices, reports, computer abstracts, and other documentation. The pertinent references are listed at the end of this report. Additional information was obtained by telephone contacts and a local visit with universities and private industry. These contacts are also listed.

The survey identified 23 CAI systems for which some data were available and which were not primarily experimental and limited. These CAI systems were:

APL	ELIZA	PICLS	SCHOLAR-TEACH
CD/TS	FOIL	PIL	SIMON
CHIMP	LOGO	PLANIT	STRINGCOMP
CLIC	LYRIC	PLATO	TICCIT
COPI	MENTOR	RASCAL	WRITEACOURSE
COURSEWRITER	PCDP	SCHOLAR	

Analysis of CAI Systems

Comparison of CAI Systems Against DEVTOS Requirements. The DEVTOS requirements to be met by a CAI system, as specified in TM-5076/001/00, were:

1. Is the system programmed in FORTRAN or CDC 3000 COMPASS language?

2. Does the system operate on, or can it be modified to operate on the CDC 3300 DEVTOS system? This requirement encompasses similarity of machines on which the systems are operational, upward and downward compatibility within a manufactured line, second versus third generation hardware and also the aspect of system design for machine transferability.

3. Does the system require less than 81,000 (24 bit) words of Core? This requirement is dictated by the maximum core size of the DEVTOS computer.

4. Does the system require less than 20 million bytes of Auxiliary storage? This requirement is dictated by the DEVTOS maximum auxiliary storage.

5. Is the system able to interface to at least 12 CRT displays on-line? The desired medium of student interface to the system is with CRT display terminals. Each of 10 students will require a CRT terminal. Two additional CRT terminals were used by project members for monitor and control.

6. Does system interface with Selectric typewriters? A requirement exists for making hardcopy of various data.

7. Can the system operate under the control of a time-sharing system? The DEVTOS operating system requires that a time-sharing capability exist.

8. Does the system have adequate documentation for identifying and programming proper interfaces to DEVTOS? This requirement is crucial in the interface programming due to the limited time available to perform the programming.

9. Does the system process a standard character set identified for DEVTOS? The DEVTOS system can handle only a subset of the characters generally available on other systems. Some of the characters are further restricted in DEVTOS.

10. Is the system in the public domain?

Some of these requirements were immediately disqualifying; the first requirement, "Is the system programmed in FORTRAN or the CDC 3300 COMPASS language?" was one of these. Six of the CAI languages qualified: CHIMP, CLIC, FOIL, LYRIC, PLANIT, and WRITEACOURSE. Discussions with the developers of WRITEACOURSE at the University of Washington indicated that it was not currently used there nor known to be used elsewhere. Consequently, WRITEACOURSE was dropped.

The remaining five CAI systems were compared against all the requirements. The results are shown in Figure 6.

Explanation of entries for DEVTOS requirements 2 and 8 of Figure 6:

Item 2

   1 = extreme difficulty in modifying the systems is predicted.

   2 = modification would be difficult in that documentation is a
       list of the program and handwritten notes (FOIL); or
       documentation about the language is good but the state of
       documentation on the system is not known (CHIMP).

   4 = proven transferability and operability on CDC 3000 series
       machines.

<u>Item 8</u>

 0 = no documentation

 2 = some documentation but lacks direct interface points

 3 = documentation is adequate but could use more detail

 4 = complete documentation exists

Figure 6 indicates that CHIMP and PLANIT ranked above the other three systems in meeting the DEVTOS requirements.

DEVTOS REQUIREMENTS	CHIMP	CLIC	FOIL	LYRIC	PLANIT
1. FORTRAN	Yes	Yes	Yes	Yes	Yes
2. CDC 3300	2	1	2	1	4
3. CORE	Yes	Yes	Yes	Yes	Yes
4. AUX	Yes	Yes	Yes	Yes	Yes
5. CRT	Yes	Yes	Yes	Yes	Yes
6. SELECTRIC	Yes	Yes	Yes	Yes	Yes
7. OP SYSTEM	Yes	Yes	Yes	Yes	Yes
8. DOCUMENTATION	3	0	2	4	4
9. CHARACTER	Yes	Yes	Yes	Yes	Yes
10. PUBLIC	No	Yes	Yes	No	Yes

Figure 6. Comparison of CAI systems against DEVTOS AI requirements

<u>Comparison of CAI Systems Against Minimum Elements Required</u>. The five remaining CAI systems were then compared against the minimum essential elements considered necessary for a CAI system to function in practice. There were 39 of these elements derived essentially from a list of over 90 items compiled by Zinn.[5,6] These 39 elements were determined to be the basic components needed for a CAI system to be used in an experimental environment. These 39 elements were separated into seven groups of related elements and a matrix was prepared for each group. A "yes" entry

5 Zinn, K. L. <u>Comparative study of languages for programming interactive use of computers in instruction</u>. Boston, Mass: Educom, 1969.

6 Zinn, K. L. <u>An evaluative review of users of computers in instruction Project Clue (Computer Learning Under Evaluation)</u>. Ann Arbor, Mich.: University of Michigan, Center for Research on Learning and Teaching, December 1970.

indicates the element exists and a "no" entry that it does not. Some
elements receive a qualified "yes" or "no" which is indicated by an
"*" followed by an explanation.

MATRIX I.   GENERAL CHARACTERISTICS ELEMENTS

1) Can a lesson contain at least 100 ''frames''?*

2) Can sufficient text be presented at one time so as to fill a 20-line,
50-character DEVTOS CRT?

3) Can lessons be loaded off-line in card image input?

4) Is the length of an acceptable answer from the student at least one
full line?

5) Can the system give the student multi-lined feedback?

Element	CHIMP	CLIC	FOIL	LYRIC	PLANIT
1. Length	Yes	Yes	Yes	Yes	Yes
2. Test	Yes	Yes	Yes	Yes	Yes
3. Card Input	Yes	Yes	Yes	Yes**	Yes
4. Answer	Yes	Yes	Yes	Yes	Yes
5. Feedback	Yes	Yes	Yes	Yes	Yes

---

*A frame is the basic building block of lessons. Some languages do not
call their units or segments frames, but it is the most commonly used
term in AI.

**System documentation mentions paper tape as the only off-line input.
However, this was in reference to the system being attached to a terminal
using paper tape. The paper tape is in card-image format, hence this
tends to indicate that card or card image information may be handled
by the LYRIC system with minor adaptation.

MATRIX II. LESSON EXECUTION ELEMENTS

1) Can the student be presented with only textual material that does not require a response to question?

2) Can the system handle a constructed response type of answer from the student in which the student types a free form or sentence type answer?

3) Is the system capable of a multiple choice type question?

   (Elements 4, 5, and 6 all pertain to the capability of the system to make decisions based on past performance by the student. Each asks the question, ''Are the proper tools for this task available?'')

4) Does the language contain the following decision connectives: IF, AND, OR?

5) Does the language have at least the following or equivalent set of logical operators: EQUAL, LESS THAN, GREATER THAN, LESS THAN or EQUAL TO, GREATER THAN or EQUAL TO?

6) Can decisions be based on every frame within the lesson?

7) Can the student leave in the middle of the lesson and know where he is?

8) On a student's returning from a lesson which he did not complete, is the student automatically restarted at the point at which he stopped?

9) Is there an automatic review by topic function that can be manipulated by the author?

Element	CHIMP	CLIC	FOIL	LYRIC	PLANIT
1. Text only	Yes	Yes	Yes	Yes	Yes
2. Constructed	Yes	Yes	Yes	Yes	Yes
3. Multiple Choice	Yes	Yes	Yes	Yes	Yes
4. Connectives	Yes	Yes	Yes	Yes	Yes
5. Operators	Yes	Yes	Yes	Yes	Yes
6. Decisions	Yes	Yes	Yes	Yes	Yes
7. Leave	Yes	Yes	Yes	Yes	Yes
8. Restart	No*	Yes	No*	No*	Yes
9. Review	No	No	No	No	Yes

---

*The student must know and supply the proper frame which he was in when he left.

MATRIX III.   LESSON BUILDING ELEMENTS

1) Does the language have a facility to give a lesson a name and execute the lesson by some command containing the name as a component of the command?

2) Can frames be labeled with a meaningful alphanumeric mnemonic?

3) Is there a designator used for the correct answer?

4) Is there a designator used for the incorrect answer?

5) Does the language associate a tag with each answer or otherwise allow for several answers with different feedback associated with individual answers?

6) Is there a facility for handling unanticipated answers?

Element	CHIMP	CLIC	FOIL	LYRIC	PLANIT
1. Name	Yes	Yes	Yes	Yes	Yes
2. Labels	Yes	Yes	Yes	No	Yes
3. Correct	Yes	Yes	Yes	Yes	Yes
4. Incorrect	Yes	Yes	Yes	Yes	Yes
5. Answer tags	Yes	Yes	Yes	Yes	Yes
6. Unanticipated	Yes	Yes	Yes	Yes	Yes

MATRIX IV. COUNTERS ELEMENTS (NOTE: These are items that may be arithmet-
ically set to values by the author)

1) Are there at least 20 counters available?

2) Are the legal values for the counters of a significantly large range?

3) Can the following common arithmetic operators be used with the counters:
   / (divide); ** (exponent); + (addition); - (subtraction); * (multipli-
   cation)?

4) Can the counters be fully interrogated through the use of decision
   statements?

5) Can the author print the contents of the counters?

6) Does the system have counters for the time of day?

Element	CHIMP	CLIC	FOIL	LYRIC	PLANIT
1. Number	Yes	Yes	Yes	Yes	Yes
2. Values	Yes	Yes	Yes	Yes	Yes
3. Operators	Yes	Yes	Yes	Yes	Yes
4. Interrogation	Yes	Yes	Yes	Yes	Yes
5. Print	Yes	Yes	Yes	Yes	Yes
6. Time	Yes	No	Yes	No	Yes

MATRIX V. ANSWER PROCESSING SERVICE FUNCTIONS ELEMENTS

1) Does the system have the capability for allowing misspellings of an answer?

2) Will the system allow for extraneous words in an answer?

3) Will punctuation be ignored?

4) Can a partial answer be acceptable and recorded?

5) Can the order of a list as an answer be variable and still correct?

Element	CHIMP	CLIC	FOIL	LYRIC	PLANIT
1. Misspelling	No	No	No	No	Yes
2. Extraneous	Yes	Yes	Yes	Yes	Yes
3. Punctuation	Yes	Yes	No	Yes	Yes
4. Partial	Yes	Yes	Yes	Yes	Yes
5. Order	Yes	No	Yes	No	Yes

MATRIX VI. BRANCHING ELEMENTS

1) Can the system allow branches to frame numbers?

2) Can the system allow branches to frame labels?

3) Can the system allow branches to another lesson?

4) Can the system branch back to the same point in a lesson it has branched from?

Element	CHIMP	CLIC	FOIL	LYRIC	PLANIT
1. Numbers	Yes	Yes	Yes	Yes	Yes
2. Labels	Yes	Yes	Yes	No	Yes
3. Lesson	No	Yes	No	No	Yes
4. Back	No	No	No	No	Yes

MATRIX VII. AUTOMATIC STUDENT RECORD KEEPING ELEMENTS

(NOTE: The experimental design required that the following data be available for analysis: frame number or identifier; whether the frame was answered correctly or incorrectly; the student latency time for the frame; the exact anticipated answer matched. Only PLANIT and CHIMP record these items automatically. In the three other systems this is done by the use of individual counters which can be cumbersome.)

1) Is each frame the student sees recorded?

2) Is the fact that the student answered correct or incorrect recorded?

3) Is the latency time for the frame recorded?

4) Is the exact anticipated answer matched recorded?

Element	CHIMP	CLIC	FOIL	LYRIC	PLANIT
1. Number	Yes	No*	No*	No*	Yes
2. Right/wrong	Yes	No*	No*	No*	Yes
3. Latency	Yes	No	Yes	No	Yes
4. Exact answer	Yes	No*	No*	No*	Yes

---

*This could be accomplished by the use of counters.

Results of Analysis. The pertinent factors and conclusions derived from the analysis are summarized for each CAI system as follows:

1. **CLIC**. CLIC was developed in the environment of UT-2D operating system (CDC 6500) and avails itself of special system interfaces. Internal documentation is practically non-existent. There is also no facility for recording latency. Consequently, CLIC did not meet the requirements.

2. **LYRIC**. LYRIC is a proprietary product with a single copy price of $12,600. LYRIC has no facility for recording latency. LYRIC is written to interface to an interactive FORTRAN which may be difficult to use in the DEVTOS batch FORTRAN mode. Consequently, LYRIC did not meet the requirements.

3. **FOIL**. The FOIL system is lacking in the areas of answer processors and automated record keeping. The system has run only on IBM/360 machines and it appears that the only interface docmuentation is a commented listing of the program. Consequently, FOIL did not meet the requirements.

4. **CHIMP**. The CHIMP system is capable of handling all the requirements. Dr. Munn has stated that there should be no problems in converting CHIMP to another system. Although special care may have been taken in the use of FORTRAN, CHIMP has run only on the UNIVAC 1108 which is a third generation machine and the CDC 3300 in DEVTOS is really a second generation machine. The conversion factors of CHIMP are unknown in both time and materials. However, the input/output of the two machines would be different. Furthermore, the interface requirements are not documented. It is only known to be specially tailored for the 1108 and can be assumed to take advantage of the 1108's third generation capabilities. CHIMP is a proprietray product of the University of Maryland. Discussions with Dr. Munn, the developer of CHIMP, indicated that action by the university board of trustees would be needed to release the system and that a price or royalty would be involved. Whether the system would be released or what time and cost would be involved was not known. In spite of these factors, CHIMP is considered an excellent system. Consequently, CHIMP did meet the requirements except for the unknowns discussed above.

5. **PLANIT**. PLANIT meets every matrix element and has excellent interface documentation. At the time of the survey it was being implemented on a CDC 3170 at California State University at Northridge. The CDC 3170 is very similar to the CDC 3300. Additionally, it had been implemented on numerous other computers. The PLANIT CAI system tapes are evidently available from a number of possible sources. Consequently, PLANIT does meet the requirements.

Conclusions from Survey

Two acceptable CAI systems, CHIMP and PLANIT, met the requirements for implementation on DEVTOS. Certain aspects of PLANIT were still being debugged; however, these were not considered disqualifying. The status of PLANIT at the time of the survey is discussed in the following section of this report.

Based upon the analysis conducted, PLANIT met the project requirements best and was recommended for implementation on DEVTOS.

Status of PLANIT at the time of the Survey

There were several sources for PLANIT systems, and several PLANIT installations. There was also a PLANIT users group chaired by Dr. Mort Rahimi, Computer Science Department, Michigan State University, which published a newsletter.

Sources for PLANIT systems:
1) Control Data Corporation
2) Michigan State University
3) National Science Foundation
4) System Development Corporation

Some PLANIT installations:
1) California State University at Northridge California (on CDC 3170)
2) Katholieke University, Nijmegen, Netherlands (on IBM 370/155)
3) Michigan State University (on CDC 6500)
4) Purdue University (on CDC 6500)
5) University of Freiburg, West Germany (on Siemens 4004/45)

PLANIT was originally written by SDC and operated successfully on an IBM Q-32 computer. The system was written in JOVIAL and interfaced to a time-sharing operating system. This system utilized teletypes, Cathode Ray Tubes (CRT) and RAND tablets as input/output devices. In 1968, SDC received a contract from the National Science Foundation to develop a prototype of a machine transferable version of PLANIT, so that the system could interface to any medium-to-large scale computer operating in either batch or time-sharing operating systems. In December of 1970 this work was completed. At that time, PLANIT was demonstrated to be machine transferable; it was transferred from a time-shared XDS 940 computer with a 24-bit word and 8-bit byte to a 360/40 running under DOS, a batch operating system, with a 32-bit word and 8-bit byte. Also at that time, the basic system was demonstrated to be operable. However, work on PLANIT was stopped as there was no money available for quality-assurance or field testing. Consequently, this version of PLANIT had not been completely debugged and there were problems with it.

Both SDC and Control Data Corporation had been working on debugging PLANIT. The exact status of the CDC PLANIT was unavailable, but sources reported that their newest version was fairly clean. At that time, CDC considered its version of PLANIT to be proprietary at a cost of $15,000, including installation.

The SDC version of PLANIT had three major bugs remaining. The first involved an "unnamed" lesson in the PLANIT lesson building mode, which could be worked around by the lesson designer. The second limited an author's use of the extensive PLANIT decision language, which would restrict him somewhat in writing lessons. The third problem was that student records of PLANIT were allocated improperly, so that several different students' records became jumbled into a single student record. For analysis purposes this was unacceptable.

Early in 1971, the University of Freiburg, Freiburg, West Germany, had begun an installation and shakedown of the PLANIT system. SDC furnished some program support and the PLANIT tape which had met the acceptance tests on SDC's Santa Monica computer but required considerable time and effort before it became operational on the Freiburg computer. Dr. Charles Frye, acting as a private consultant, was in Freiburg connected with this effort. Upon his return to the United States near the end of 1972, Dr. Frye continued his PLANIT work at Michigan State University and subsequently delivered a copy of the PLANIT that he was using to the National Science Foundation. The National Science Foundation has given a grant to Purdue University to complete a field test of PLANIT. Dr. Frye has been given an NSF grant to correct the remaining problems and update the documentation. This effort was expected to be completed sometime before June 1973. SDC had been informed by Dr. Frye of the following problems that still existed but were to be corrected:

1. SAVE and GET of lessons from/to tape, which would not affect Army Tactical Data System for Training (ATDST).

2. UNLOAD student records to tape (would not affect ATDST).

3. Pattern matching in PLANIT decision language (could affect ATDST, see PLANIT decision language above).

There were two feasible options for ARI to obtain operational copies of a debugged PLANIT for use in DEVTOS.

The first was through the National Science Foundation. The NSF PLANIT was scheduled for release by June 1973, or before. The NSF version was expected to be error free. In case a few errors still existed, they would be identified and documented. The contact was:

Mr. Erik D. McWilliams
Program Director
Computer Technology and Systems
Office of Computing Activities
National Science Foundation
Washington, D.C. 20550
Phone: (202)-282-7935

The second option was that Michigan State University and Control Data Corporation, under a teaming arrangement, had been working on PLANIT for a period of time, Control Data Corporation had indicated that they could supply the system, support it and guarantee installation on the CDC 3300 for $15,000 to $20,000. The contact was:

Mr. Dan Burgess
CDC Computer Systems Division
2200 North Berkshire Lane
Minneapolis, Minnesota
Phone: (612) 545-2851

In the Interim, SCD supplied a version of PLANIT (and an accompanying listing) which was not completely debugged but could be used to determine the DEVTOS interface problems and programming required.

CAI Systems

APL   (A Programming Language). A scientific-mathematical
                  language first developed by IBM. It is an interactive
                  language with several facilities for CAI. Although the
                  language has been implemented by several other computer
                  vendors, the language does not operate on the CDC 3300.
                  Orange Coast and Golden West Junior Colleges in Orange
                  County, California have used the language extensively
                  for CAI.

                  Contact: Coast Community College District
                            Office of Educational Development
                            1370 Adams Avenue
                            Costa Mesa, Calif. 92626
                            Dr. Bernard S. Luskin, Director

                                      and

                            IBM Corporation
                            Thomas J. Watson Research Center
                            P. O. Box 218
                            Yorktown Heights, New York 10598
                            Dr. E. N. Adams, Director

CD/TS   (Computer-Directed Training Subsystem). A CAI system, written
             in COBOL for the Burroughs 3500, which is a subset of PLANIT.
             Developed by SDC for the U. S. Air Force.

                    Contact: Command Systems Division
                               Electronic Systems Division
                               Air Force Systems Command
                               U. S. Air Force

CHIMP   CAI system developed by Institute for Molecular Physics,
             University of Maryland. System is owned by the University
             and any release would have to be approved by Board of
             Trustees. System is written in FORTRAN for the UNIVAC 1108.

                    Contact: Robert J. Munn
                               Professor of Chemistry
                               Department of Chemistry
                               University of Maryland
                               College Park, Maryland

CLIC   (Conversational Language for Instructional Computing). A
             multipurpose CAI language written in FORTRAN for the CDC 6500
             operating under UT-2D operating system at the University of
             Texas. Although the system was once written to be trans-
             ferable, the present system ''has been developed in the
             environment of out own (Univ. of Texas) UT-2D operating system
             and avails itself of special system interfaces,...Internal
             documentation is practically nonexistent.''

                    Contact: Edwin P. Shaw, Assistant Director
                               Computation Center
                               University of Texas at Austin
                               Austin, Texas   78712

COPI   (Computer-Oriented Programmed Instruction). A system written
             and marketed by UNIVAC for use on the UNIVAC 1108.   The
             system is written in UNIVAC Assembly language. This system is
             currently being used by the Marine Corps at 29 Palms, Calif.

                    Contact: UNIVAC
                               1333 Camino Adel Rio South
                               San Diego, California
                               Mr. Ken Corbett

COURSEWRITER   The first author-language for CAI, it was developed by IBM
               and written in machine language. The most used CAI language
               in the world. The system operates all through the U.S. and
               Europe. A COURSEWRITER system has been in use for several
               years at Fort Monmouth.

                         Contact: IBM Corporation
                                    Thomas J. Watson Research Center
                                    P.O. Box 218
                                    Yorktown Heights, N.Y.   10598

ELIZA   A language developed by MIT for the IBM 7090 written in the
               LISP language. The language has interesting capabilities for
               simulated dialogue.

                         Contact: Educational Research Center
                                    Massachusetts Institute of Technology
                                    Cambridge, Mass.

FOIL   (File Oriented Interpretive Language). Developed at the
               Center for Research on Learning and Teaching, University of
               Michigan.   Written in IBM FORTRAN IV.

                         Contact: Karl Zinn
                                    C R L T
                                    University of Michigan
                                    Ann Arbor, Michigan

LOGO - MENTOR - SCHOLAR - SIMON - STRINGCOMP

               These five systems are all special experimental CAI languages
               developed at Bolt, Beranak, and Newman. All are written in
               various special-purpose languages.

                         Contact: Bolt, Beranak, and Newman, Inc.
                                    Educational Technology Department
                                    50 Moulton Street
                                    Cambridge, Mass.   02138
                                    Wallace Feurzeig, Director

LYRIC   (Language for Your Remote Instruction by Computer). A
                 fully conversational CAI language developed, trade-marked,
                 sold and marketed by CAIS.

                           Contact: Computer-Assisted Instruction Systems
                           979 Teakwood Road
                           Los Angeles, California 90049
                           Dr. Gloria Silvern Altschiller

PCDP   (Physics Computer Development Project). An excellent
                 language specifically suited to Physics and other hard
                 sciences with excellent use of graphics. System written
                 in metasymbol for the XDS Sigma-7.

                           Contact: Alfred Bork
                           PCDP
                           University of California at Irvine
                           Irvine, California 92664

PICLS   (Purdue Instructional and Computational Learning System).
                 Developed at Purdue University by A. Oldehoeft for the CDC
                 6500 in machine language. Language no longer used.

PIL   (Pittsburgh Interpretive Language). A CAI system written in
                 assembly language for the PDP-10. Also in assembly language
                 for the IBM 360 and 370.

                           Contact: Computing Center
                           University of Pittsburgh
                           Pittsburgh, Pennsylvania

PLANIT   (Programming Language for Interactive Teaching). Originally
                 written in JOVIAL for the IBM ANSO-32 by SDC. It was
                 rewritten in FORTRAN and designed to be machine transferable
                 under a NSF grant. PLANIT is generally recognized as the
                 most complete and versatile of the CAI author languages.

                           Contact: System Development Corporation
                           PLANIT
                           2500 Colorado Avenue
                           Santa Monica, California 90406

PLATO   (Programmed Logic for Automatic Teaching Operation). This system is now in its fourth generation. Its present language is called TUTOR which is written in CDC 6000 assembly language. PLATO consists of several special pieces of hardware among them the PLATO IV Plasma Display Terminal

> Contact: Dr. Don Bitzer, Director
> Computer-Based Educational Research
> Laboratory
> University of Illinois
> Urbana, Illinois

RASCAL   (Rudimentary Adaptive System for Computer-Aided Language). Produced as part of a Master's thesis by John Christopher Stewart, Lt., U.S. Navy for a Master of Science in Computer Science from the Naval Postgraduate School, Monterey, California. The system is written in PL/1.

> Contact: Naval Postgraduate School
> Monterey, California

SCHOLAR-TEACH   An elementary CAI system written in machine language for the DEC-System 10. Originally written by Boeing, Seattle, Washington.

> Contact: Digital Equipment Corporation
> Educational Products Group
> Maynard, Mass. 01754

TICCIT   (Timed-Shared, Interactive, Computer-Controlled Information Television). TICCIT is a specialized CAI system using mini-computers, cable television, and color television with keyboards as input/output devices. The hardware system is being developed by the Mitre Corporation. Courseware is being developed by Dr. Victor Bundersen of Brigham Young University. The system will have its initial testing early in 1974.

> Contact: Mitre Corporation
> 1820 Dolly Madison Blvd.
> McLean, Virginia 22101
> Kenneth J. Stetten, Principal Investigator

WRITEACOURSE   Originally programmed in PL/1 and more recently in FORTRAN.
                The University of Washington people indicate it is not now
                being used there and they also are not aware of any use of
                the system being made elsewhere.

                        Contact: Dr. Earl Hunt
                                   University of Washington
                                   Department of Psychology
                                   Seattle, Washington

SUMMARY

Twenty-three CAI systems were identified and analyzed. The charac-
teristics of these systems were compared against the specific require-
ments which must be met for a system to operate in DEVTOS. Of the 23,
six met the major requirement that they be written in either the COMPASS
or FORTRAN programming language. One system, although written in FORTRAN,
was eliminated because it had no known use at that time. The five CAI
systems remaining were analyzed against the other DEVTOS requirements and
also against the minimum essential elements considered necessary for a
CAI system to be practically functional. By the time these requirements
and elements had been considered, two CAI systems--CHIMP and PLANIT--
remained. Both of these met the CAI requirements and elements specified.
There were no known drawbacks to implementing either of these on DEVTOS.
However, PLANIT--as compared to CHIMP--had a known transferability record,
was currently on a CDC computer, and had many additional capabilities in
regard to course development and record keeping. Further, many univer-
sities were using PLANIT for course development and the possibility of
using these courses on tactical computers was attractive.

For the reasons stated in this report, PLANIT was clearly identified
as the CAI system recommended for implementation in DEVTOS.

SOURCES OF INFORMATION

<u>PERSONNEL CONTACTED</u>:

Mr. David Aaronson	SDC
Mr. J. Alexander	TSDG
Mr. Jim Baker	ARI
Mr. L. A. Bowman	BRC
Major Burgess	TSDG
Mr. Stu Charlston	SDC
Lt. Col. Ezikiel	TSDG, Executive Officer
Mr. Norman Fredericks	Computer Science Corporation
Mr. George Gividen	ARI, Head Ft. Hood Field Unit
Dr. Gloria Grace	SDC
Major Buzz Hensel	TSDG, Head AI Project Committee
Dr. William Hoyt	SDC
Col. Kuhlman	TSDG, Commander
T. Sgt. Meriweither	TSDG
Major Miller	ARTADS Representative at TSDG, Member AI Project Committee
Dr. Brian Murphy	SDC
Mr. John Noyes	BRC, Team Chief
Dr. Charles Nystrom	ARI, Fort Hood, Member AI Project Committee
Lt. Col. Palmer	TSDG
Mr. Jim Pessin	BRC
T. Sgt. Salsbury	TSDG
Mr. Mike Shaw	TSDG, Member AI Project Committee
Major Curt Tomlin	TSDG
Mr. Frank Urffer	TSDG
Mr. Jim Westwick	TSDG
Dr. Henry Zagorski	SDC
Lt. Col. Rose	ARTADS LNO, MASSTER
Lt. Col. Andre	USACSC
Mr. W. Elliot	USACSC
Mr. R. Marion	ARI

DOCUMENTATION SOURCES

A. United States Army Computer Systems Command, TSDG Documents:

1. Technical Documentation Report, United States Army Computer Systems Command 01-04-02 Tactical Operations System, Design Specifications, Drum Retrofit Appendix A, TOS Operating System Logic Design. Bunker Ramo Corporation Technical Staff for U. S. Army Computer Systems Command, June 1972.

2. Tactical Operations System Programmer Reference Manual. Bunker Ramo Corporation, June 1972.

3. System Operations Manual for Developmental Tactical Operations System (DEVTOS). Bunker Ramo Corporation, April 1972.

4. Tactical Operations System Design Specifications, Basic Document. Bunker Ramo Corporation, Fort Belvoir, Virginia, May 1969.

5. Staff Operating Procedures Manual for Developmental Tactical Operations System (DEVTOS). Bunker Ramo Corporation, Fort Hood, Texas, March 1973.

6. Introduction and Staff Management Procedures for Developmental Tactical Operations System (DEVTOS). Bunker Ramo Corporation, Fort Hood, Texas, March 1972.

7. Tactical Operations System (TOS) Software System Description. Major Curtis S. Tomlin, USACSC-TSDG, Systems Analysis Branch, Fort Hood, Texas, January 1972.

B. Bunker Ramo Corporation Documents:

1. Tactical Operations System. Bunker Ramo Corporation - Electronic Systems Division - DEVTOS.

C. Control Data Corporation (CDC) documents:

1. 3290-4 (Edit) Multistation Terminal Operating and Programming Guide Pub. No. PD 82142900, May 1969.

2. 216 Remote Terminal Operating and Programming Guide Pub. No. PD 82147900, December 1969.

3. 3300/3500 Mass Storage FORTRAN/Master Programming Guide Pub. No. 60178400, January 1967.

4. 3100 3200 3300 Computer Systems FORTRAN General Information Manual Pub. No. 60055200, Rev. C, September 1968.

5. 3300 3500 Computer Systems USASI FORTRAN/ Master General Information Manual Pub. No. 60206700A, February 1968.

Preceding page blank

6. **3100 3200 3300 3500 Computer System FORTRAN Reference Manual** Pub. No. 60057600, Rev. B.   August 1966.

7. **3100 3200 3300 Computer Systems FORTRAN Library Functions** Pub. No. 60138600.   **August 1966.**

8. **Time-Sharing Equipment Bulletin** Pub. No. 60156000.   November 1965.

9. **3300 3500 Computer Systems Master General Information Manual** Pub. No. 60134100.   October 1965.

10. **Business Data Processing Equipment Bulletin**   Pub. No. 60157500. November 1965.

11. **3300/3500 Compass/Master** Pub. No. 60178500.   December 1966.

12. **3000L PERT/COST/MASTER**   Pub. No. 60179800.   May 1967.

13. **3000L PERT/TIME/MASTER**   Pub. No. 60179700.   May 1967.

14. **3200 Control Data Computer System Generalized, Sort/Merge Program, General Information Manual** Pub. No. 60055800, Rev. A. May 1964.

15. **3100 3200 3300 Computer Systems Report Generator General Information Manual**   Pub. No. 60133100.   August 1965.

16. **3100 3200 3300 Computer Systems Adapt General Information Manual** Pub. No. 60135900.   April 1966.

17. **3100 3200 3300 3500 Computer Systems ALGOL General Information Manual** Pub. No. 60173100.   August 1966.

18. **3500 Computer System Respond/MSOS General Information Manual** Pub. No. 60190400.   March 1967.

19. **3100 3200 3300 3500 Computer Systems Saint General Information Manual** Pub. No. 60132100, Rev. A. November 1964.

20. **3100 3200 3300 3500 Computer System Compatible Compass Language Reference Manual**   Pub. No. 60174000. August 1966.

21. **Time Sharing Business Processor Guide Control Data 3300/3500.** October 1965.

22. **Time-Sharing a Management Look at Control Data 3300/3500.** October 1965.

23. **A Report on Time-Sharing for Business and Industry with the Control Data 3300/3500.**   November 1965.

24. 3100 3200 3300 **Computer Systems Autoload Utility Reference Manual** Pub. No. 60130100. January 1965

25. 3100 3200 **Computer Systems Scope/Disk Scope Reference Manual** Pub. No. 60171200. June 1966.

26. **Programming Reference Aids** Pub. No. 60158600. April 1966.

27. 3100 3200 3300 **Computer Systems Library Routines** Pub. No. 60058100. January 1965.

28. 3100 3200 3300 3500 **Computer Systems Data Processing Package Reference Manual** Pub. No. 60059300, Rev. A. April 1966.

29. 3300 3500 **Computer Systems Mass Storage Sort/Master Reference Manual** Pub. No. 60206600. July 1967.

30. 3300 3500 **Computer Systems Master Operator's Guide** Pub. No. 60178600. April 1967.

31. 3300 **Computer System Respond/MSOS Reference Manual** Pub. No. 60192800. September 1967.

32. 3300/3500 **Master Product Set Diagnostic Handbook** Pub. No. 60206800. September 1967.

33. 3300 3500 **Computer Systems Mass Storage COBOL/Master Reference Manual** Pub. No. 60192000. July 1967.

BIBLIOGRAPHY

Altschiller, G. S. LYRIC User Manual. Los Angeles: Computer-Assisted Instructional Systems, December 1968.

Altschiller, Dr. Gloria Silvern. Personal communication, 1973.

Auner, R. A. TUTOR User's Memo. Urbana, Ill: University of Illinois Computer-based Educational Research Laboratory, August 1971.

Auner, R. A. TUTOR User's Memo. Urbana, Ill.: University of Illinois Computer-based Educational Research Laboratory, January 1971 (revised April 1972).

Badger, George, University of Pittsburgh Computing Center. Personal communication, 1973.

Bennik, F. D., and Frye, C. H. PLANIT Author's Guide. Santa Monica, Calif.: System Development Corporation, TM(L)-4422/001/01, October 1, 1970.

Bitzer, D., and Kaperoas, D. S. The Design of an Economically Viable Large-Scale Computer Based Education System. Urbana, Ill.: University of Illinois, January 1972.

Bohn, R. E. TUTOR User's Memo TUTOR Commands. Urbana, Ill.: University of Illinois, Computer-Based Educational Research Laboratory, July 1971 (revised January 1972).

Bork, A. M. and Mosmann, C. Teaching Conversations with the XDS SIGMA 7. Irvine, Calif.: University of California, Physics Computer Development Project, January 1971.

Burgess, Dan, Control Data Corporation, Personal Communication, 1973.

Butler, A. K. and Frye, C. H. PLANIT Language Reference Manual, System Development Corporation, TM(L)-4422-002/01, Santa Monica, Calif: October 1, 1970.

Carbonell, J. R. Artificial Intelligence and Large Interactive Man Computer Systems, Cambridge, Mass.: Bolt, Beranak and Newman, Inc., November 1970.

Carbonell, J. R., et al. Mixed-Initiation Systems for Training and Decision-Aid Applications. Cambridge, Mass.: Bolt, Beranak and Newman, Inc., November 1970.

Engel, G. L. Computer Assisted Instruction: A Selected Bibliography and KWIC Index. Dahlgren, Va.: U. S. Naval Weapons Laboratory, April 1969.

Preceding page blank

Feurzeig, W. Computer Systems for Teaching Complex Concepts. Cambridge, Mass.: Bolt, Beranak and Newman, Inc., March 1969.

Frye, C. H. NW Regional Education Laboratory. Personal Communication, 1973.

Hartman, S. A., et al. CLIC Users Manual. University of Texas at Austin, Computation Center.

Hessebart, J. C., et al. File-Oriented Interpretive Language, Part II: A Manual for Programmers. Ann Arbor, Mich.: University of Michigan, Center for Research on Learning and Teaching, April 1969.

Hunt, E. University of Washington, Department of Psychology, Personal communication, 1973.

Hunt, E., and Zosel, M. WRITEACOURSE: An Educational Programming Language. Proceeding FJCC, 1968.

Kaken, H. A. (Ed.). Index to Computer-Assisted Instruction (third ed.). New York: Harcourt Brace Jovanovich, Inc., 1971.

Kraatz, J. TUTOR User's Memo Introduction to TUTOR. Urbana, Ill.: University of Illinois, Computerized-Based Educational Research Laboratory, June 1972.

Longo, A. A. Implementation of Computer Assisted Instruction in U.S. Army Basic Electronic Training, Follow-up of a Feasibility Study. Fort Monmouth, N.J.: U.S. Army Signal Center and School, September 1969.

MITRE Corporation. Toward a Market Success for CAI: An Overview of the TICCIT Program. McLean, Va.: MITRE Corp., June 1972.

Munn, R. J. CHIMP Author's Manual, College Park Md.: University of Maryland, Institute for Molecular Physics, 1969.

Munn, R. J., University of Maryland, Institute for Molecular Physics. Personal Communication, 1973.

Myers, D., University of Washington, Computer Science Laboratory, Personal Communication, 1973.

Noe, Dr., Chairman, University of Washington Computer Science Group. Personal Communication, 1973.

Rahimi, Dr. M., Michigan State University. Personal communication, 1973.

Shaw, E. P., Assistant Director, Computation Center, University of Texas at Austin. Personal communication, 1973.

Silvern, L. Personal communication, 1973.

Special Interest Group in Computer Uses in Education. SIGCUE Bulletin.
Vols. 5-7). New York: Association of Computing Machinery, 196 -7 .

Stegelman, R. California State University at Northridge, Personal
Communication. 1 .

Stewart, J. C. RASCAL: A Rudimentary Adaptive System for Computer-
Aided Learning. Monterey, Calif: Naval-Post Graduate School, Master's
Thesis, December 1 0.

Stifle, J. The PLATO IV Student Terminal. Urbana, Ill.: University of
Illinois, Computer-Based Educational Research Laboratory, March 1 .
a

Stifle, J. The PLATO IV Architecture. Urbana, Ill.: University of
Illinois, Computer-Based Educational Research Laboratory, May 1972. (b)

Stifle, J., Bitzer, D., and Johnson, M. Digital Data Transmission via
CATV. Urbana, Ill.: University of Illinois, Computer-Based Educational
Research Laboratory, May 1 .

Tucker, P. T. Large Scale Computer Terminal Output Controller. Urbana,
Ill.: University of Illinois, Computer-Based Educational Research
Laboratory, June 1 1.

Urion, H. L. Cum-Laude: A New Approach in Designing Computer Assisted
Language Problems. Hanover, N.H.: Thayer School of Engineering,
September 1 0.

Walker, G. S., et al. Appilcation of Computers in Education and Training
Systems: A Survey of Computer Assisted Instructional Centers. Brooks
AFB, Tex.: Air Force Human Resource Laboratory, December 1 0.

Zinn, K. L. Comparative Study of Languages for Programming Interactive
Use of Computers in Instruction. Boston, Mass: Educom, 1 6 .

Zinn, K. L. An Evaluative Review of Users of Computers in Instruction
Project Clue (Computer Learning Under Evaluation). Ann Arbor, Mich.:
University of Michigan, The Center for Research on Learning and
Teaching, December 19 0.

GLOSSARY

Glossary of abbreviations and acronyms is supplied to aid the reader.

AI	Automated Instruction (used interchangeably with (CAI))
ARI	Army Research Institute
ARTADS	Army Tactical Data Systems
ATDST	Army Tactical Data System for Training
BRC	Bunker-Ramo Corporation
CAI	Computer Aided Instruction (used interchangeably with AI)
CCC	Central Computer Complex
CDC	Control Data Corporation
CRT	Cathode Ray Tube
DEVTOS	Developmental Tactical Operations System
IBM	International Business Machines Corporation
I/O	Input/Output
MASSTER	Modern Army Selected Systems Test Evaluation and Review
MIOP	Machine Interface I/O Program
MSU	Michigan State University
NSF	National Science Foundation
OCF	Operator Communication Field
PLANIT	Programming Language for Interactive Teaching
RSDT	Remote Station Data Terminal
SDC	System Development Corporation
SPR	Special Process Request messages
SRI	Standing Request for Information
TOC	Tactical Operations Center
TOS	Tactical Operations System
TSDG	Tactical Systems Development Group
UCLA	University of California at Los Angeles
UIOD	User Input Output Device
USACSC	U.S. Army Computer Systems Command

Preceding page blank

CHARACTER SET

(a)	A-Z	Letters		(m)	)	Right parenthesis
(b)	0-9	Numbers		(n)	;	Semicolon
(c)		Space		(o)	/	Slash
(d)	.	Period		(p)	:	Colon
(e)	,	Comma		(q)	@	At
(f)	-	Hyphen (or minus)		(r)	!	Exclamation
(g)	+	Plus		(s)	?	Question
(h)	=	Equals		(t)	"	Quotes
(i)	%	Percent		(u)	#	Number
(j)	*	Asterisk		(v)	<	Less than
(k)	$	Dollar		(w)	>	Greater than
(l)	(	Left parenthesis		(x)	'	Prime
				(y)	&	And